


```

LL               IIIIII               SSSSSSSS
LL               IIIIII               SSSSSSSS
LL               II                  SS
LL               II                  SS
LL               II                  SS
LL               II                  SS
LL               II                  SSSSSS
LL               II                  SSSSSS
LL               II                  SS
LL               II                  SS
LL               II                  SS
LL               II                  SS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```

```
1 0001 0 %TITLE 'File operations for footnote processing'
2 0002 0 MODULE foofil ( IDENT = 'V04-000'
3 P 0003 0 %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
4 0004 0 NONEXTERNAL = LONG_RELATIVE)]
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
33 0033 1
34 0034 1 ABSTRACT:
35 0035 1
36 0036 1 Handle all file operations for .FOOTNOTE and .END FOOTNOTE
37 0037 1 commands.
38 0038 1
39 0039 1 ENVIRONMENT: Transportable
40 0040 1
41 0041 1 AUTHOR: R.W.Friday CREATION DATE: September, 1978
42 0042 1
43 0043 1
```

FOOFIL
V04-000

File operations for footnote processing
Revision History

F 10
16-Sep-1984 00:31:31
14-Sep-1984 13:06:23

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]FOOFIL.BLI;1

Page 2
(2)

:	45	0044	1	%SBTTL 'Revision History'
:	46	0045	1	!
:	47	0046	1	! MODIFIED BY:
:	48	0047	1	!
:	49	0048	1	! 009 RER00009 Ron Randall 07-Mar-1983
:	50	0049	1	! Global edit of all modules. Updated module names, idents,
:	51	0050	1	! copyright dates. Changed require files to BLISS library.
:	52	0051	1	! !--
:	53	0052	1	!
:	54	0053	1	!

```
56 0054 1 %SBTTL 'Module Level Declarations'
57 0055 1
58 0056 1
59 0057 1 : TABLE OF CONTENTS:
60 0058 1
61 0059 1 FORWARD ROUTINE
62 0060 1   FOOFIL;
63 0061 1
64 0062 1
65 0063 1 : INCLUDE FILES:
66 0064 1
67 0065 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
68 0066 1 REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
69 0197 1
70 U 0198 1 %IF DSRPLUS %THEN
71 U 0199 1 LIBRARY 'REQ:DPLLIB';           ! DSRPLUS BLISS Library
72 0200 1 %ELSE
73 0201 1 LIBRARY 'REQ:DSRLIB';           ! DSR BLISS Library
74 0202 1 %FI
75 0203 1
76 0204 1
77 0205 1 : MACROS:
78 0206 1
79 0207 1 MACRO
80 M 0208 1   erm_t (rnfcod, str_descr) =
81 M 0209 1     BEGIN
82 M 0210 1     BIND
83 M 0211 1       temp = str_descr : $STR_DESCRIPTOR ();
84 M 0212 1
85 M 0213 1     erme (rnfcod, .temp [STR$A_POINTER], .temp [STR$H_LENGTH], .semcod)
86 0214 1     END   %;
87 0215 1
88 0216 1
89 0217 1 : EXTERNAL REFERENCES:
90 0218 1
91 0219 1 EXTERNAL
92 0220 1   FNCT : FNCT_DEFINITION,           ! Overall footnote statistics.
93 0221 1   FNESIZ : FN_EXT_SIZE_DEFINITION,   ! Size of each pending footnote.
94 0222 1   FNISIZ : FN_INT_SIZE_DEFINITION,   ! Internal size of each pending
95 0223 1                                     ! footnote (number of TSF/MRA pairs)
96 0224 1   MRA : REF FIXED STRING,
97 0225 1   TSF : TSF_DEFINITION;
98 0226 1
99 0227 1 EXTERNAL
100 0228 1   FO1IOB : $XPO_IOB (),              ! Only the address is looked at.
101 0229 1   FO0IOB : REF -$XPO_IOB (),         ! The primary footnote IOB. Except for
102 0230 1                                     ! FOO_CLEAN, all footnote operations
103 0231 1                                     ! refer to this IOB.
104 0232 1   FOTIOB : REF $XPO_IOB (),          ! Used as a work file for footnotes.
105 0233 1   FFNAME : $STR_DESCRIPTOR (CLASS = DYNAMIC), ! Failure filename destination
106 0234 1   SEMCOD,                             ! Secondary error message code
107 0235 1   FOOREC : FOOREC_DEFINITION,       ! Work area for pass-thru records.
108 0236 1   FOMRA : FIXED STRING,             ! Substitute MRA.
109 0237 1   FOOTSF : VECTOR [TSF_SIZE];        ! Substitute TSF area
110 0238 1
111 0239 1 EXTERNAL LITERAL
112 0240 1   RNFCOF,                             ! Error messages
```

FOOFIL
V04-000

File operations for footnote processing
Module Level Declarations

H 10
16-Sep-1984 00:31:31
14-Sep-1984 13:06:23

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]FOOFIL.BLI;1

Page 4
(3)

:	113	0241	1	RNFCRF,
:	114	0242	1	RNFCWF,
:	115	0243	1	RNFILE;
:	116	0244	1	
:	117	0245	1	EXTERNAL ROUTINE
:	118	0246	1	ERM,
:	119	0247	1	ERMÉ,
:	120	0248	1	ERMS,
:	121	0249	1	GRAB_RESULTANT;
:	122	0250	1	

```
124 0251 1 %SBTTL 'body of routine'
125 0252 1 GLOBAL ROUTINE foofil (operation) =
126 0253 1
127 0254 1 ++
128 0255 1 FUNCTIONAL DESCRIPTION:
129 0256 1
130 0257 1     Performs operations on the footnote file.
131 0258 1
132 0259 1 FORMAL PARAMETERS:
133 0260 1
134 0261 1     OPERATION indicates what should be done.  See the file FFDEFS.REQ.
135 0262 1
136 0263 1 IMPLICIT INPUTS:      None
137 0264 1
138 0265 1 IMPLICIT OUTPUTS:     None
139 0266 1
140 0267 1 ROUTINE VALUE:
141 0268 1 COMPLETION CODES:      None
142 0269 1
143 0270 1 SIDE EFFECTS:         None
144 0271 1 --
145 0272 1
146 0273 2 BEGIN
147 0274 2 LOCAL
148 0275 2 STATUS;
149 0276 2
150 0277 2 CASE .OPERATION FROM 0 TO FFOO_OP OF
151 0278 2 SET
152 0279 2
153 0280 2 [FOO_INIT] :                ! Initialize footnote file.
154 0281 2 BEGIN
155 0282 2 ! When initializing the IOB make sure the associated file name is
156 0283 2 ! correct.
157 0284 2 IF .FOOIOB EQLA F01IOB
158 0285 2 THEN
159 P 0286 2 $XPO_IOB_INIT ( IOB = .FOOIOB,
160 P 0287 2 FILE_SPEC = ('001RNO.TMP'),
161 0288 2 ATTRIBUTES = BINARY)
162 0289 2 ELSE
163 P 0290 2 $XPO_IOB_INIT ( IOB = .FOOIOB,
164 P 0291 2 FILE_SPEC = ('002RNO.TMP'),
165 0292 2 ATTRIBUTES = BINARY);
166 0293 2 RETURN FOO_NORMAL;
167 0294 2 END;
168 0295 2
169 0296 2 [FOO_OPIN] :                ! Open footnote file for input.
170 0297 2 BEGIN
171 0298 2
172 0299 2 IF .FOOIOB [IOB$V_OPEN]    ! Is the file open?
173 0300 2 THEN
174 0301 2 IF .FOOIOB [IOB$V_INPUT] ! Yes, but is it for input?
175 0302 2 THEN
176 0303 2 RETURN FOO_NORMAL    ! Yes again, so return.
177 0304 2 ELSE
178 0305 2 ! NO, it's open for output.  We can't have that!  We must
179 0306 2 ! close it and reopen it for input.
180 0307 2 BEGIN
```

```

181      0308 4      foofil (foo_clos);      ! Always succeeds, so no status
182      0309 4      ! check needed.
183      0310 4
184      P 0311 4      status = $XPO_OPEN ( IOB = .foojob
185      P 0312 4      ,OPTIONS = INPUT
186      P 0313 4      ,FAILURE = grab_resultant
187      0314 4      );
188      0315 4
189      0316 4      IF .status
190      0317 4      THEN
191      0318 4      RETURN foo_normal
192      0319 4      ELSE      !Can't open footnote file for reading.
193      0320 5      BEGIN
194      0321 5      erm t (rnfcdf, ffname);
195      0322 5      RETURN foo_bad;
196      0323 4      END;
197      0324 4
198      0325 4      END
199      0326 3      ELSE
200      0327 3      ! The file is not open at all. This indicates some serious
201      0328 3      error, since the only reason the file could be closed is
202      0329 3      either that the user has defined no footnotes, or all the
203      0330 3      footnotes have already been output. At any rate, output
204      0331 3      an error message and return.
205      0332 4      BEGIN
206      0333 4      ERMS (RNFILE, CH$PTR (UPLIT ('FOO_OPIN')), 9);
207      0334 4      RETURN FOO_BAD
208      0335 4      END
209      0336 4
210      0337 2      END;
211      0338 2
212      0339 2      [FOO_RHDR] :
213      0340 2      ! Read a record header and copy the information into a safe place.
214      0341 3      BEGIN
215      P 0342 3      STATUS = $XPO_GET (IOB = .FOOJOB,
216      P 0343 3      FAILURE = XPO$IO FAILURE,
217      0344 3      FULLWORDS = FOOREC_SIZE);
218      0345 3
219      0346 3      IF NOT .STATUS      ! Make sure the read went ok
220      0347 3      THEN
221      0348 3
222      0349 3      IF .STATUS EQL XPO$_END_FILE
223      0350 3      THEN      ! End of file is ok. Just means there's
224      0351 3      RETURN FOO_ENDFIL ! nothing more to read.
225      0352 3      ELSE      ! Couldn't read the record header.
226      0353 4      BEGIN
227      0354 4      ERM (RNFCRF, 0, 0);
228      0355 5      RETURN (FOO_BAD)
229      0356 3      END;
230      0357 3
231      0358 3      ! The record header has been read ok. Now copy
232      0359 4      ! the information where it will be safe.
233      0360 4      BEGIN
234      0361 4      BIND
235      0362 4      RECORD_HEADER = FOOJOB [IOB$_DATA] : REF VECTOR [FOOREC_SIZE];
236      0363 4
237      0364 4      INCR I FROM 0 TO (FOOREC_SIZE - 1) DO
      FOOREC [I] = .RECORD_HEADER [I];

```

```
238 0365 4
239 0366
240 0367
241 0368 ! Signal successful completion to the user.
242 0369 RETURN FOO_NORMAL
243 0370
244 0371
245 0372
246 0373 [FOO_OPOU] :
247 0374 ! Open footnote file for output.
248 0375 BEGIN
249 0376
250 0377 ! If the file is already opened for output just return.
251 0378 IF .FOOIOB [IOBSV_OPEN]
252 0379 THEN ! The file is at least open.
253 0380
254 0381 IF .FOOIOB [IOBSV_OUTPUT] ! See if it's open for output.
255 0382 THEN ! The file is in exactly the state we want it in.
256 0383 RETURN FOO_NORMAL
257 0384 ELSE ! The file is open for input, so we have to clean it up
258 0385 RETURN FOOFIL (FOO_CLEAN);
259 0386
260 0387 ! The file is not open at all. So start anew.
261 0388 ! Initialize the footnote IOB.
262 0389 FOOFIL (FOO_INIT);
263 0390
264 0391 !Open file for output now.
265 0392
266 0393 status = $XPO_OPEN ( IOB = .fooiob
267 0394 ,OPTIONS = OUTPUT
268 0395 ,FAILURE = grab_resultant
269 0396 );
270 0397
271 0398 IF .status
272 0399 THEN
273 0400 RETURN foo_normal
274 0401 ELSE ! Can't open footnote file for output.
275 0402 BEGIN
276 0403 erm t (rnfcof, fname);
277 0404 RETURN foo_bad;
278 0405 END;
279 0406
280 0407 END;
281 0408
282 0409 [FOO_READ] :
283 0410 ! Read back a TSF/MRA pair. A basic assumption is that the user
284 0411 ! has already done a FOO_RHDR and found out that that is indeed
285 0412 ! what's coming, and that FOOREC_RECORD_SIZE contains the length of
286 0413 ! the TSF (which was fetched automatically by the FOO_RHDR
287 0414 ! operation). After the record header that was read, comes the
288 0415 ! actual TSF. Then comes a record header for the MRA, and then
289 0416 ! comes the actual MRA.
290 0417 BEGIN
291 0418 !First, get the description of the text that has been saved.
292 0419 STATUS = $XPO_GET ( IOB = .FOOIOB, !
293 0420 FAILURE = XPOSIO_FAILURE,
294 0421 FULLWORDS = .FOOREC_RECORD_SIZE);
```

```
295 0422
296 0423
297 0424
298 0425
299 0426
300 0427
301 0428
302 0429
303 0430
304 0431
305 0432
306 0433
307 0434
308 0435
309 0436
310 0437
311 0438
312 0439
313 0440
314 0441
315 0442
316 0443
317 0444
318 0445
319 0446
320 0447
321 0448
322 0449
323 0450
324 0451
325 0452
326 0453
327 0454
328 0455
329 0456
330 0457
331 0458
332 0459
333 P 0460
334 P 0461
335 0462
336 0463
337 0464
338 0465
339 0466
340 0467
341 0468
342 0469
343 0470
344 0471
345 0472
346 0473
347 0474
348 0475
349 0476
350 0477
351 0478

IF NOT .STATUS
THEN
    IF .STATUS EQL XPOS_END_FILE
    THEN RETURN FOO_ENDFIL ! Reached the end of the footnote file.
    ELSE ! Couldn't read back footnote record.
        BEGIN
            ERM (RNFCRF, 0, 0);
            RETURN FOO_BAD;
        END;

! The text descriptor (TSF) was read correctly. Now move the
! results into a waiting TSF. The move is done because the next
! read invalidates the current IOB pointers.
BEGIN
    BIND
        FOOTSF = FOOIOB [IOB$A_DATA] : REF VECTOR;

! Note that FOOREC_RECORD_SIZE amount of information is copied
! instead of TSF_SIZE amount of information. That allows other
! routines to write only a part of a TSF, if desired.
INCR I FROM 0 TO .FOOREC_RECORD_SIZE - 1 DO
    TSF [I] = .FOOTSF [I];

END;

! Get the record header for the MRA that must follow.
! Note that this might be a good place to put a consistency check.
FOOFIL (FOO_RHDR);

! Now, get the actual saved text. Note that FULLWORDS could be
! computed as CH$ALLOCATION (.TSF_INT_HL + .TSF_INT_VL + 1) but
! it's not, since that information is available in the record
! header. Also, that allows other code to write more or less
! information, if desired.
STATUS = $XPO_GET ( IOB = .FOOIOB,
                    FAILURE = XPOS_IO_FAILURE,
                    FULLWORDS = .FOOREC_RECORD_SIZE);

! Make sure the read went ok.
IF NOT .STATUS
THEN ! Couldn't read back MRA.
    BEGIN
        ERM (RNFCRF, 0, 0);
        RETURN FOO_BAD;
    END;

! Data was read, so make MRA look like a real FIXED string.
FS_START (MRA) = CH$PTR (.FOOIOB [IOB$A_DATA]);
FS_NEXT (MRA) = .FS_START (MRA);
FS_MAXSIZE (MRA) = .TSF_INT_HL + .TSF_INT_VL;
FS_LENGTH (MRA) = .TSF_INT_HL + .TSF_INT_VL;
RETURN FOO_NORMAL;
END;
```

```

352 0479 2
353 0480
354 0481 [FOO_PREAD] :
355 0482   ! Read a record that's by definition a "pass through" record.
356 0483   ! Implicit in this processing is that the user has already read the
357 0484   ! record header in FOOREC and determined that indeed, a "pass
358 0485   ! through" record is to be read. The "pass through" record is not
359 0486   ! copied anywhere, but merely left in the buffer, where XPORT as
360 0487   ! put it. The address is returned in FOOREC_ADDRESS.
361 P 0488   BEGIN
362 P 0489   STATUS = $XPO_GET (IOB = .FOOIOB,
363 0490   FAILURE = XPOSIO FAILURE,
364 0491   FULLWORDS = .FOOREC_RECORD_SIZE);
365 0492
366 0493   ! Make sure the read went ok.
367 0494   IF NOT .STATUS
368 0495   THEN
369 0496   BEGIN
370 0497   ERM (RNFCRF, 0, 0);
371 0498   RETURN FOO_BAD
372 0499   END;
373 0500   ! The read went ok. Let the user know where the data is.
374 0501   FOOREC_ADDRESS = .FOOIOB [IOB$A_DATA];
375 0502
376 0503   ! Tell the user everything went ok.
377 0504   RETURN FOO_NORMAL
378 0505   END;
379 0506
380 0507 [FOO_WRITE] :
381 0508   ! Write a footnote record. A footnote record consists of a TSF
382 0509   ! record and an MRA record. record header preceeds each record.
383 0510   ! Note that this code leaves the minor record type (FOOREC_MINOR_CODE)
384 0511   ! alone, so that the user can put whatever he wants into it.
385 0512   BEGIN
386 0513
387 0514   ! First create the record header for the TSF
388 0515   FOOREC_MAJOR_TYPE = FOOREC_MAJ_TSF;           ! Major record type
389 0516   FOOREC_RECORD_SIZE = TSF_SIZE;               ! Length of the TSF
390 0517
391 0518   ! Now actually write out the record header
392 P 0519   STATUS = $XPO_PUT ( IOB = .FOOIOB,
393 P 0520   FAILURE = XPOSIO FAILURE,
394 0521   BINARY_DATA = (FOOREC_SIZE, FOOREC));
395 0522
396 0523   IF NOT .STATUS
397 0524   THEN
398 0525   BEGIN
399 0526   ERM (RNFCWF, .STATUS, 0);
400 0527   RETURN FOO_BAD
401 0528   END;
402 0529
403 P 0530   STATUS = $XPO_PUT ( IOB = .FOOIOB, ! Write the real TSF
404 P 0531   FAILURE = XPOSIO FAILURE,
405 0532   BINARY_DATA = (TSF_SIZE, .TSF) );
406 0533
407 0534   IF NOT .STATUS
408 0535   THEN
409 0536   BEGIN
410 0537   ERM (RNFCWF, .STATUS, 0);
411 0538   RETURN FOO_BAD
412 0539   END;
413 0540   ! If we couldn't write the TSF,
```

```

: 409      0536      4      BEGIN
: 410      0537      4      ERM (RNFCWF, .STATUS, 0);      ! then
: 411      0538      4      RETURN FOO_BAD;      ! report the error
: 412      0539      4      END;      ! and exit
: 413      0540      4
: 414      0541      4      ! Create the record header for the MRA
: 415      0542      4      FOOREC_MAJOR_TYPE = FOOREC_MAJ_MRA;
: 416      0543      4      FOOREC_RECORD_SIZE = CH$ALLOCATION ( .TSF_INT_HL + ! The number
: 417      0544      4      ! .TSF_INT_VL + ! of fullwords
: 418      0545      4      ! );      ! (at least 1)
: 419      0546      4
: 420      0547      4      ! And write it out.
: 421      P 0548      4      STATUS = $XPO_PUT ( IOB = .FOOIOB,
: 422      P 0549      4      FAILURE = XPOSIO_FAILURE,
: 423      0550      4      BINARY_DATA = (FOOREC_SIZE, FOOREC));
: 424      0551      4
: 425      0552      4      ! Make sure the write went ok.
: 426      0553      4      IF NOT .STATUS
: 427      0554      4      THEN
: 428      0555      4      BEGIN
: 429      0556      4      ERM (RNFCWF, .STATUS, 0);
: 430      0557      4      RETURN FOO_BAD
: 431      0558      4      END;
: 432      0559      4
: 433      0560      4      ! Write the MRA. Always write at least 1 word.
: 434      P 0561      4      STATUS = $XPO_PUT ( IOB = .FOOIOB,
: 435      P 0562      4      FAILURE = XPOSIO_FAILURE,
: 436      P 0563      4      BINARY_DATA = ( .FOOREC_RECORD_SIZE,
: 437      0564      4      MRA [5] );
: 438      0565      4
: 439      0566      4      !Update current size of this footnote as it's being output.
: 440      0567      4      FNESIZ [.FNCT_COUNT] = .FNESIZ [.FNCT_COUNT] + .TSF_LINES;
: 441      0568      4
: 442      0569      4      IF .STATUS
: 443      0570      4      THEN
: 444      0571      4      BEGIN      ! If the above write went OK, update the
: 445      0572      4      ! current internal size of this footnote
: 446      0573      4      ! (number of TSF/MRA pairs)
: 447      0574      4      FNISIZ [.FNCT_COUNT] = .FNISIZ [.FNCT_COUNT] + 1;
: 448      0575      4      RETURN FOO_NORMAL;      ! and exit
: 449      0576      4      END
: 450      0577      4      ELSE      ! If we couldn't write the MRA.
: 451      0578      4      BEGIN
: 452      0579      4      ERM (RNFCWF, .STATUS, 0);      ! report the error
: 453      0580      4      RETURN FOO_BAD;      ! and exit
: 454      0581      4      END;
: 455      0582      4      END;
: 456      0583      4
: 457      0584      4      [FOO_PWRITE] :      ! Write a "pass through" record to the footnote
: 458      0585      4      BEGIN      ! temporary file:
: 459      0586      4
: 460      0587      4      ! We assume that FOOREC_MINOR_TYPE has already been set,
: 461      0588      4      ! so we don't change it here.
: 462      0589      4
: 463      0590      4      FOOREC_MAJOR_TYPE = FOOREC_MAJ_PASS; ! Set the major record type.
: 464      0591      4
: 465      P 0592      4      STATUS = $XPO_PUT (IOB = .FOOIOB,      ! Write the record header.
```

```
466 P 0593 3 FAILURE = XPOSIO FAILURE,
467 0594 3 BINARY_DATA = (FOOREC_SIZE, FOOREC));
468 0595 3
469 0596 3 IF NOT .STATUS
470 0597 3 THEN
471 0598 3 BEGIN
472 0599 3 ERM (RNFCWF, .STATUS, 0);
473 0600 3 RETURN FOO_BAD;
474 0601 3 END;
475 0602 3
476 0603 3 ! Now write the actual "pass through" record
477 P 0604 3 STATUS = $XPO_PUT (IOB = .FOOIOB,
478 P 0605 3 FAILURE = XPOSIO FAILURE,
479 0606 3 BINARY_DATA = (.FOOREC_RECORD_SIZE, .FOOREC_ADDRESS));
480 0607 3
481 0608 3 IF NOT .STATUS
482 0609 3 THEN
483 0610 3 BEGIN
484 0611 3 ERM (RNFCWF, .STATUS, 0);
485 0612 3 RETURN FOO_BAD;
486 0613 3 END;
487 0614 3
488 0615 3 ! Update the record count for this footnote
489 0616 3 FNISIZ [.FNCT_COUNT] = .FNISIZ [.FNCT_COUNT] + 1;
490 0617 3
491 0618 3 ! Tell the user everything went ok.
492 0619 3 RETURN FOO_NORMAL
493 0620 3 END;
494 0621 3
495 0622 3 [FOO_CLOSE] :
496 0623 3 BEGIN
497 P 0624 3 $XPO_CLOSE (IOB = .FOOIOB,
498 0625 3 OPTIONS = REMEMBER);
499 0626 3
500 0627 3 RETURN FOO_NORMAL;
501 0628 3 END;
502 0629 3
503 0630 3 [FOO_DELE] :
504 0631 3
505 0632 3 ! Delete the footnote file.
506 0633 3 BEGIN
507 0634 3 $XPO_DELETE (IOB = .FOOIOB, FAILURE = 0);
508 0635 3 RETURN FOO_NORMAL;
509 0636 3 END;
510 0637 3
511 0638 3 [FOO_CLEAN] :
512 0639 3 ! The footnote file is open for reading, but we want to put a new
513 0640 3 ! footnote at the end of it. Therefore, copy the remainder of the
514 0641 3 ! footnote file into a new file, and leave it open for output, so
515 0642 3 ! that stuff can be written to the end of the file.
516 0643 3 BEGIN
517 0644 3 $XPO_IOB_INIT (IOB = .fotioB);
518 0645 3
519 0646 3 ! Open the work file. Note that its name depends on which IOB
520 0647 3 ! FOTIOB refers to.
521 0648 3 IF .FOTIOB EQLA F01IOB
522 0649 3 THEN
```

```
523      status = $XPO_OPEN ( IOB = .fotioB
524      P 0651      ,OPTIONS = OUTPUT
525      P 0652      ,ATTRIBUTES = BINARY
526      P 0653      ,FILE_SPEC = ('001RNO.TMP')
527      P 0654      ,FAILURE = grab_resultant
528      P 0655      )
529      ELSE
530      P 0657      status = $XPO_OPEN ( IOB = .fotioB
531      P 0658      ,OPTIONS = OUTPUT
532      P 0659      ,ATTRIBUTES = BINARY
533      P 0660      ,FILE_SPEC = ('002RNO.TMP')
534      P 0661      ,FAILURE = grab_resultant
535      P 0662      );
536      P 0663
537      P 0664      ! Verify that this work file could be opened.
538      P 0665      IF NOT .status
539      P 0666      THEN
540      P 0667      ! Something went wrong
541      P 0668      BEGIN
542      P 0669      erm t (rnfcwf, fname);
543      P 0670      RETURN foo_bad
544      P 0671      END;
545      P 0672
546      P 0673      ! Now copy everything from the active footnote file into
547      P 0674      ! the footnote work file.
548      P 0675      STATUS = XPOS_NORMAL;
549      P 0676      WHILE .STATUS NEQ XPOS_END_FILE DO
550      P 0677      BEGIN
551      P 0678      ! First read the record header that describes each record
552      P 0679      STATUS = $XPO_GET (IOB = .FOOIOB,
553      P 0680      FAILURE = XPOS_IO_FAILURE,
554      P 0681      FULLWORDS = FOOREC_SIZE);
555      P 0682
556      P 0683      IF NOT .STATUS
557      P 0684      THEN
558      P 0685      ! Something's not quite right. See what it is.
559      P 0686      IF .STATUS EQL XPOS_END_FILE
560      P 0687      THEN
561      P 0688      ! Get out of this copy loop if nothing more to read.
562      P 0689      EXITLOOP
563      P 0690      ELSE
564      P 0691      ! Something serious. Can't read back a footnote record.
565      P 0692      BEGIN
566      P 0693      ERM (RNFCRF, 0, 0);
567      P 0694      RETURN FOO_BAD
568      P 0695      END;
569      P 0696
570      P 0697      ! The read went correctly. Now simply copy it to the work file.
571      P 0698      STATUS = $XPO_PUT (IOB = .FOTIOB, BINARY_DATA = FOOIOB [IOBST_DATA]);
572      P 0699
573      P 0700      ! Check that it went ok.
574      P 0701      IF NOT .STATUS
575      P 0702      THEN
576      P 0703      ! Something happened.
577      P 0704      BEGIN
578      P 0705      ERM (RNFCWF, .STATUS, 0);
579      P 0706
```

```

580      0707 5      RETURN FOO_BAD
581      0708 4      END;
582      0709 4
583      0710 4      ! Read the record that follows the record header.
584      0711 5      BEGIN
585      0712 5      BIND
586      0713 5      FOOREC = FOOIOB [IOB$A_DATA] : REF FOOREC_DEFINITION;
587      0714 5
588      0715 5      ! Read precisely as much as the record header says was written.
589      P 0716 5      STATUS = $XPO_GET (IOB = .FOOIOB,
590      P 0717 5      FAILURE = XPO$IO_FAILURE,
591      0718 5      FULLWORDS = .FOOREC_RECORD_SIZE);
592      0719 5
593      0720 5      ! Make sure the read went ok.
594      0721 5      IF NOT .STATUS
595      0722 5      THEN
596      0723 5      ! Something went wrong. End of file is not allowed here,
597      0724 6      BEGIN ! so it must be an error.
598      0725 6      ERM (RNFCRF, 0, 0);
599      0726 7      RETURN (FOO_BAD)
600      0727 5      END;
601      0728 5
602      0729 5      ! Now write this information to the work file.
603      P 0730 5      STATUS = $XPO_PUT (IOB = .FOTIOB,
604      0731 5      BINARY_DATA = FOOIOB [IOB$T_DATA]);
605      0732 5
606      0733 5      ! Make sure the write went ok.
607      0734 5      IF NOT .STATUS
608      0735 5      THEN
609      0736 5      ! Something bad happened
610      0737 6      BEGIN
611      0738 6      ERM (RNFCWF, .STATUS, 0);
612      0739 6      RETURN FOO_BAD
613      0740 5      END;
614      0741 5
615      0742 4      END;
616      0743 4
617      0744 3      END; ! End of copying from FOOIOB to FOTIOB.
618      0745 3
619      0746 3      ! Get rid of the file that we just copied stuff from.
620      P 0747 3      $XPO_CLOSE (IOB = .FOOIOB, ! Close the temp file, but
621      0748 3      OPTIONS = REMEMBER); ! remember its name.
622      P 0749 3      $XPO_DELETE (IOB = .FOOIOB, ! Now, delete it, but don't
623      0750 3      FAILURE = 0); ! barf if it isn't there.
624      0751 3
625      0752 3      ! Now switch IOB pointers because all the other FOOFIL operations
626      0753 3      ! work only on FOOIOB.
627      0754 4      BEGIN
628      0755 4      LOCAL
629      0756 4      TEMP;
630      0757 4
631      0758 4      TEMP = .FOTIOB;
632      0759 4      FOTIOB = .FOOIOB;
633      0760 4      FOOIOB = .TEMP;
634      0761 3      END;
635      0762 3
636      0763 3      !Update the footnote counting information
```

```

: 637      0764 3      INCR I FROM 0 TO .FNCT_COUNT - 1 DO
: 638      0765 4      BEGIN
: 639      0766 4      FNESIZ [.I] = .FNESIZ [.I + .FNCT_OLD];
: 640      0767 4      FNISIZ [.I] = .FNISIZ [.I + .FNCT_OLD];
: 641      0768      END;
: 642      0769      !And now forget there ever were any old footnotes.
: 643      0770      FNCT_OLD = 0;
: 644      0771      RETURN FOO_NORMAL
: 645      0772      END;
: 646      0773
: 647      0774
: 648      0775
: 649      0776      TES;
: 650      0777
: 651      0778 1      END;

```

!End of FOOFIL

```

.TITLE FOOFIL File operations for footnote processing
.IDENT \V04-000\

```

```

.PSECT $PLITS$,NOWRT,NOEXE,2

```

```

50 4D 54 2E 4F 4E 52 31 30 30 00000 P.AAA: .ASCII \001RNO.TMP\
50 4D 54 2E 4F 4E 52 32 30 30 0000A P.AAB: .ASCII \002RNO.TMP\
      4E 49 50 4F 5F 4F 4F 46 00014 P.AAC: .ASCII \FOO OPIN\
50 4D 54 2E 4F 4E 52 31 30 30 0001C P.AAD: .ASCII \001RNO.TMP\
50 4D 54 2E 4F 4E 52 32 30 30 00026 P.AAE: .ASCII \002RNO.TMP\

```

```

.PSECT $OWNS$,NOEXE,2

```

```

000A 00000 $IOB$FILE SPEC:
      .WORD 10
01 0E 00002 .BYTE 14, 1
00000000' 00004 .ADDRESS P.AAA
000A 00008 $IOB$FILE SPEC:
      .WORD 10
01 0E 0000A .BYTE 14, 1
00000000' 0000C .ADDRESS P.AAB
000A 00010 $IOB$FILE SPEC:
      .WORD 10
01 0E 00012 .BYTE 14, 1
00000000' 00014 .ADDRESS P.AAD
000A 00018 $IOB$FILE SPEC:
      .WORD 10
01 0E 0001A .BYTE 14, 1
00000000' 0001C .ADDRESS P.AAE

```

```

.EXTRN FNCT, FNESIZ, FNISIZ
.EXTRN MRA, TSF, FOTIOB
.EXTRN FOOIOB, FOTIOB, FFNAME
.EXTRN SEMCOD, FOOREC, FOOMRA
.EXTRN FOOTSF, RNFCOF, RNFCRF
.EXTRN RNFCWF, RNFILE, ERM
.EXTRN ERME, ERMS, GRAB, RESULTANT
.EXTRN XPOSOPEN, XPOSGET
.EXTRN XPOSIO_FAILURE, XPOSPUT
.EXTRN XPOSCLUSE, XPOSFAILURE

```

.EXTRN XPOSDELETE

.PSECT \$CODE\$,NOWRT,2

.ENTRY FOOFIL, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
R11

0252

MOVAB XPOSGET, R11

MOVAB XPOSPUT, R10

MOVAB XPOSIO_FAILURE, R9

MOVAB FOORECT8, R8

MOVAB FOOI0B, R7

SUBL2 #8, SP

CASEL OPERATION, #0, #10

0643

.WORD 2\$-1\$,-

6\$-1\$,-

12\$-1\$,-

14\$-1\$,-

24\$-1\$,-

29\$-1\$,-

30\$-1\$,-

32\$-1\$,-

9\$-1\$,-

20\$-1\$,-

27\$-1\$,-

MOVL FOOI0B, R6

MOVAB FOIIOB, R0

CML R6, R0

BNEQ 3\$

MOVCS #0, (SP), #0, #244, (R6)

0284

0288

MOVL #50397245, (R6)

MOVAB \$IOB\$FILE_SPEC, 4(R6)

BRB 4\$

MOVCS #0, (SP), #0, #244, (R6)

0292

MOVL #50397245, (R6)

MOVAB \$IOB\$FILE_SPEC, 4(R6)

MOVW #526, 30(R6)

BISB2 #1, 48(R6)

BRW 47\$

MOVL FOOI0B, R0

BLBC 50(R0), 8\$

BLBS 46(R0), 5\$

PUSHL #5

CALLS #1, FOOFIL

MOVL FOOI0B, R0

BISB2 #1, 46(R0)

MOVB #1, 44(R0)

PUSHAB GRAB_RESULTANT

CLRL -(SP)

PUSHL R0

CALLS #3, XPOSOPEN

MOVL R0, STATUS

BLBS STATUS, 5\$

BRW 35\$

PUSHL #9

PUSHAB P.AAC

0643

0299

0301

0308

0314

0316

0321

0333

00F4 8F 00

00F4 8F 00

04

04

1E

30

FF5F

2E

2C

00000000G

00000000'

OFFC 00000

5B 00000000G EF 9E 00002

5A 00000000G EF 9E 00009

59 00000000G EF 9E 00010

58 00000000G EF 9E 00017

57 00000000G EF 9E 0001E

5E 00000000G EF 08 C2 00025

00 04 AC CF 00028

0062 0016 0002D 1\$:

0322 01D1 00035

019C 00B1 0003D

56 67 D0 00043 2\$:

50 00000000G EF 9E 00046

50 56 D1 0004D

6E 19 12 00050

66 00 2C 00052

66 00059

04 66 0301003D 8F D0 0005A

A6 00000000' EF 9E 00061

17 11 00069

6E 00 2C 0006B 3\$:

66 00072

04 66 0301003D 8F D0 00073

A6 00000000' EF 9E 0007A

A6 020E 8F B0 00082 4\$:

01 88 00088

04AC 31 0008C 5\$:

50 67 D0 0008F 6\$:

30 A0 E9 00092

F2 2E A0 E8 00096

05 DD 0009A

CF 01 FB 0009C

50 67 D0 000A1

A0 01 88 000A4

A0 01 90 000A8 7\$:

00000000G EF 9F 000AC

7E D4 000B2

50 DD 000B4

03 FB 000B6

52 D0 000BD

C9 52 E8 000C0

0319 31 000C3

09 DD 000C6 8\$:

00000000' EF 9F 000C8

00000000G	EF	00000000G	8F	DD	000CE	PUSHL	#RNF FILE		
			03	FB	000D4	CALLS	#3, ERMS		
			03E5	31	000DB	BRW	43\$		0334
	50		67	D0	000DE	9\$:	MOVL	FOOIOB, R0	0344
34	A0		10	B0	000E1		MOVW	#16, 52(R0)	
36	A0		02	90	000E5		MOVB	#2, 54(R0)	
2C	A0		06	90	000E9		MOVB	#6, 44(R0)	
			59	DD	000ED		PUSHL	R9	
			7E	D4	000EF		CLRL	-(SP)	
			50	DD	000F1		PUSHL	R0	
	6B		03	FB	000F3		CALLS	#3, XPOSGET	
	52		50	D0	000F6		MOVL	R0, STATUS	
	58		52	E9	000F9		BLBC	STATUS, 15\$	0346
51	67		38	C1	000FC		ADDL3	#56, FOOIOB, R1	0361
			50	D4	00100		CLRL	I	0363
	FB A840		00	B140	D0 00102	10\$:	MOVL	@0(R1)[I], FOOREC[I]	0364
F5	50		03	F3	00109		AOBLEQ	#3, I, 10\$	
			042B	31	0010D	11\$:	BRW	47\$	0643
	50		67	D0	00110	12\$:	MOVL	FOOIOB, R0	0378
	OD		A0	E9	00113		BLBC	50(R0), 13\$	
F1	2E		01	E0	00117		BBS	#1, 46(R0), 11\$	0381
			07	DD	0011C		PUSHL	#7	0385
	FEDD	CF	01	FB	0011E		CALLS	#1, FOOFIL	
			04	00123		RET			
			7E	D4	00124	13\$:	CLRL	-(SP)	0389
	FED5	CF	01	FB	00126		CALLS	#1, FOOFIL	
	50		67	D0	0012B		MOVL	FOOIOB, R0	0396
	2E	A0	02	88	0012E		BISB2	#2, 46(R0)	
			FF73	31	00132		BRW	7\$	
	50		67	D0	00135	14\$:	MOVL	FOOIOB, R0	0421
34	A0		04	A5	00138		MULW3	#4, FOOREC+8, 52(R0)	
			02	90	0013D		MOVB	#2, 54(R0)	
	36	A0	06	90	00141		MOVB	#6, 44(R0)	
	2C	A0	59	DD	00145		PUSHL	R9	
			7E	D4	00147		CLRL	-(SP)	
			50	DD	00149		PUSHL	R0	
	6B		03	FB	0014B		CALLS	#3, XPOSGET	
	52		50	D0	0014E		MOVL	R0, STATUS	
	10		52	E8	00151		BLBS	STATUS, 17\$	0423
	00209000	8F	52	D1	00154	15\$:	CMPL	STATUS, #2134016	0426
			03	13	0015B		BEQL	16\$	
			0088	31	0015D		BRW	21\$	
	50		02	D0	00160	16\$:	MOVL	#2, R0	0430
			04	00163		RET			
51	67		38	C1	00164	17\$:	ADDL3	#56, FOOIOB, R1	0441
	50		01	CE	00168		MNEGL	#1, I	0447
			0A	11	0016B		BRB	19\$	
	00000000G	FF40	00	B140	D0 0016D	18\$:	MOVL	@0(R1)[I], @TSF[I]	
F2	50		68	F2	00177	19\$:	AOBLSS	FOOREC+8, I, 18\$	
			08	DD	0017B		PUSHL	#8	0453
	FE7E	CF	01	FB	0017D		CALLS	#1, FOOFIL	
	50		67	D0	00182		MOVL	FOOIOB, R0	0462
34	A0		04	A5	00185		MULW3	#4, FOOREC+8, 52(R0)	
			02	90	0018A		MOVB	#2, 54(R0)	
	36	A0	06	90	0018E		MOVB	#6, 44(R0)	
	2C	A0	59	DD	00192		PUSHL	R9	
			7E	D4	00194		CLRL	-(SP)	

			50	DD	00196	PUSHL	R0		
	6B		03	FB	00198	CALLS	#3, XPOSGET		
	52		50	DO	0019B	MOVL	R0, STATUS		
	47		52	E9	0019E	BLBC	STATUS, 21\$	0465	
	50	00000000G	EF	DO	001A1	MOVL	MRA, R0	0473	
	51		67	DO	001A8	MOVL	FOOIOB, R1		
	60	38	A1	DO	001AB	MOVL	56(R1), (R0)		
04	A0		60	DO	001AF	MOVL	(R0), 4(R0)	0474	
	51	00000000G	EF	DO	001B3	MOVL	TSF, R1	0475	
51	61	18	A1	C1	001BA	ADDL3	24(R1), (R1), R1		
08	A0		51	DO	001BF	MOVL	R1, 8(R0)		
0C	A0		51	DO	001C3	MOVL	R1, 12(R0)	0476	
			32	11	001C7	BRB	23\$	0643	
	50		67	DO	001C9	MOVL	FOOIOB, R0	0490	
34	68		04	A5	001CC	MULW3	#4, FOOREC+8, 52(R0)		
	A0		02	90	001D1	MOVB	#2, 54(R0)		
36	A0		06	90	001D5	MOVB	#6, 44(R0)		
2C			59	DD	001D9	PUSHL	R9		
			7E	D4	001DB	CLRL	-(SP)		
	6B		50	DD	001DD	PUSHL	R0		
	52		03	FB	001DF	CALLS	#3, XPOSGET		
	0B		50	DO	001E2	MOVL	R0, STATUS		
			52	E8	001E5	BLBS	STATUS, 22\$	0493	
		00000000G	7E	7C	001E8	CLRQ	-(SP)	0496	
			8F	DD	001EA	PUSHL	#RNFCRF		
	50		02C9	31	001F0	BRW	42\$		
	A8	38	67	DO	001F3	MOVL	FOOIOB, R0	0501	
08			A0	DO	001F6	MOVL	56(R0), FOOREC+16		
		F8	033D	31	001FB	BRW	47\$	0643	
	68		A8	D4	001FE	CLRL	FOOREC	0515	
	50		28	DO	00201	MOVL	#40, FOOREC+8	0516	
	6E		67	DO	00204	MOVL	FOOIOB, R0	0521	
	02		10	B0	00207	MOVW	#16, \$IOB\$OUTPUT		
03	AE		02	90	0020A	MOVB	#2, \$IOB\$OUTPUT+2		
04	AE	F8	01	90	0020E	MOVB	#1, \$IOB\$OUTPUT+3		
44	A0		A8	9E	00212	MOVAB	FOOREC, \$IOB\$OUTPUT+4		
2C	A0		6E	9E	00217	MOVAB	\$IOB\$OUTPUT, 68(R0)		
			07	90	0021B	MOVB	#7, 44(R0)		
			59	DD	0021F	PUSHL	R9		
			7E	D4	00221	CLRL	-(SP)		
	6A		50	DD	00223	PUSHL	R0		
	52		03	FB	00225	CALLS	#3, XPO\$PUT		
	6C		50	DO	00228	MOVL	R0, STATUS		
	50		52	E9	0022B	BLBC	STATUS, 25\$	0523	
	6E	A0	67	DO	0022E	MOVL	FOOIOB, R0	0532	
	02		8F	9B	00231	MOVZBW	#160, \$IOB\$OUTPUT		
03	AE		02	90	00235	MOVB	#2, \$IOB\$OUTPUT+2		
04	AE	00000000G	01	90	00239	MOVB	#1, \$IOB\$OUTPUT+3		
44	A0		EF	DO	0023D	MOVL	TSF, \$IOB\$OUTPUT+4		
2C	A0		6E	9E	00245	MOVAB	\$IOB\$OUTPUT, 68(R0)		
			07	90	00249	MOVB	#7, 44(R0)		
			59	DD	0024D	PUSHL	R9		
			7E	D4	0024F	CLRL	-(SP)		
	6A		50	DD	00251	PUSHL	R0		
	52		03	FB	00253	CALLS	#3, XPO\$PUT		
	3E		50	DO	00256	MOVL	R0, STATUS		
			52	E9	00259	BLBC	STATUS, 25\$	0534	

	F8	A8	01	DO	0025C	MOVL	#1, FOOREC	0542
		50	EF	DO	00260	MOVL	TSF, R0	0543
50		60	A0	C1	00267	ADDL3	24(R0), (R0), R0	0544
		50	04	C0	0026C	ADDL2	#4, R0	0545
68		50	04	C7	0026F	DIVL3	#4, R0, FOOREC+8	
		50	67	DO	00273	MOVL	FOOIOB, R0	0550
		6E	10	B0	00276	MOVW	#16, \$IOB\$OUTPUT	
	02	AE	02	90	00279	MOVB	#2, \$IOB\$OUTPUT+2	
	03	AE	01	90	0027D	MOVB	#1, \$IOB\$OUTPUT+3	
	04	AE	F8	A8	9E 00281	MOVAB	FOOREC, \$IOB\$OUTPUT+4	
	44	A0	6E	9E	00286	MOVAB	\$IOB\$OUTPUT, 68(R0)	
	2C	A0	07	90	0028A	MOVB	#7, 44(R0)	
			59	DD	0028E	PUSHL	R9	
			7E	D4	00290	CLRL	-(SP)	
		6A	50	DD	00292	PUSHL	R0	
		52	03	FB	00294	CALLS	#3, XPO\$PUT	
		46	50	DO	00297	MOVL	R0, STATUS	
		50	52	E9	0029A	BLBC	STATUS, 26\$	0553
		68	67	DO	0029D	MOVL	FOOIOB, R0	0564
6E		AE	04	A5	002A0	MULW3	#4, FOOREC+8, \$IOB\$OUTPUT	
	02	AE	02	90	002A4	MOVB	#2, \$IOB\$OUTPUT+2	
	03	AE	01	90	002A8	MOVB	#1, \$IOB\$OUTPUT+3	
04	AE	00000000G	EF	C1	002AC	ADDL3	#16, MRA, \$IOB\$OUTPUT+4	
		44	6E	9E	002B5	MOVAB	\$IOB\$OUTPUT, 68(R0)	
		2C	07	90	002B9	MOVB	#7, 44(R0)	
			59	DD	002BD	PUSHL	R9	
			7E	D4	002BF	CLRL	-(SP)	
		6A	50	DD	002C1	PUSHL	R0	
		52	03	FB	002C3	CALLS	#3, XPO\$PUT	
		50	50	DO	002C6	MOVL	R0, STATUS	
		50	EF	DO	002C9	MOVL	FNCT, R0	0567
		51	EF	DO	002D0	MOVL	TSF, R1	
	00000000GEF	40	A1	C0	002D7	ADDL2	52(R1), FNESIZ[R0]	
		63	52	E8	002E0	BLBS	STATUS, 28\$	0569
			01	CC	31 002E3	BRW	41\$	0578
	F8	A8	02	DO	002E6	MOVL	#2, FOOREC	0590
		50	67	DO	002EA	MOVL	FOOIOB, R0	0594
		6E	10	B0	002ED	MOVW	#16, \$IOB\$OUTPUT	
			02	90	002F0	MOVB	#2, \$IOB\$OUTPUT+2	
	02	AE	01	90	002F4	MOVB	#1, \$IOB\$OUTPUT+3	
	03	AE	F8	A8	9E 002F8	MOVAB	FOOREC, \$IOB\$OUTPUT+4	
	04	AE	6E	9E	002FD	MOVAB	\$IOB\$OUTPUT, 68(R0)	
	44	A0	07	90	00301	MOVB	#7, 44(R0)	
	2C	A0	59	DD	00305	PUSHL	R9	
			7E	D4	00307	CLRL	-(SP)	
			50	DD	00309	PUSHL	R0	
		6A	03	FB	0030B	CALLS	#3, XPO\$PUT	
		52	50	DO	0030E	MOVL	R0, STATUS	
		CF	52	E9	00311	BLBC	STATUS, 26\$	0596
		50	67	DO	00314	MOVL	FOOIOB, R0	0606
6E		68	04	A5	00317	MULW3	#4, FOOREC+8, \$IOB\$OUTPUT	
	02	AE	02	90	0031B	MOVB	#2, \$IOB\$OUTPUT+2	
	03	AE	01	90	0031F	MOVB	#1, \$IOB\$OUTPUT+3	
	04	AE	08	A8	DO 00323	MOVL	FOOREC+16, \$IOB\$OUTPUT+4	
	44	A0	6E	9E	00328	MOVAB	\$IOB\$OUTPUT, 68(R0)	
	2C	A0	07	90	0032C	MOVB	#7, 44(R0)	
			59	DD	00330	PUSHL	R9	

			7E	D4	00332	CLRL	-(SP)		
			50	DD	00334	PUSHL	R0		
	6A		03	FB	00336	CALLS	#3, XPO\$PUT		
	52		50	DO	00339	MOVL	R0, STATUS		
	A4		52	E9	0033C	BLBC	STATUS, 26\$		0608
	50	00000000G	EF	DO	0033F	MOVL	FNCT, R0		0616
		00000000GEF	40	D6	00346	INCL	FNISIZ[R0]		
			30	11	0034D	BRB	31\$		0643
	50		67	DO	0034F	MOVL	FOOIOB, R0		0625
2E	A0		10	88	00352	BISB2	#16, 46(R0)		
2C	A0		02	90	00356	MOVB	#2, 44(R0)		
		00000000G	EF	9F	0035A	PUSHAB	XPO\$FAILURE		
			7E	D4	00360	CLRL	-(SP)		
			50	DD	00362	PUSHL	R0		
00000000G	EF		03	FB	00364	CALLS	#3, XPO\$CLOSE		
			12	11	0036B	BRB	31\$		0643
	50		67	DO	0036D	MOVL	FOOIOB, R0		0634
2C	A0		03	90	00370	MOVB	#3, 44(R0)		
			7E	7C	00374	CLRL	-(SP)		
			50	DD	00376	PUSHL	R0		
00000000G	EF		03	FB	00378	CALLS	#3, XPO\$DELETE		
			01B9	31	0037F	BRW	47\$		0643
	56	00000000G	EF	DO	00382	MOVL	FOTIOB, R6		0644
00F4	8F	00	6E	00	2C	MOVCS	#0, (SP), #0, #244, (R6)		
			66		00390				
	66	0301003D	8F	DO	00391	MOVL	#50397245, (R6)		
1E	A6	020E	8F	BO	00398	MOVW	#526, 30(R6)		
	50	00000000G	EF	9E	0039E	MOVAB	FOIIOB, R0		0648
	50		56	D1	003A5	CMPL	R6, R0		
			0A	12	003A8	BNEQ	33\$		
04	A6	00000000'	EF	9E	003AA	MOVAB	\$IOB\$FILE_SPEC, 4(R6)		0655
			08	11	003B2	BRB	34\$		
04	A6	00000000'	EF	9E	003B4	MOVAB	\$IOB\$FILE_SPEC, 4(R6)		0662
2E	A6	0C010002	8F	C8	003BC	BISL2	#65538, 48(R6)		
2C	A6		01	90	003C4	MOVB	#1, 44(R6)		
		00000000G	EF	9F	003C8	PUSHAB	GRAB_RESULTANT		
			7E	D4	003CE	CLRL	-(SP)		
			56	DD	003D0	PUSHL	R6		
00000000G	EF		03	FB	003D2	CALLS	#3, XPO\$OPEN		
	52		50	DO	003D9	MOVL	R0, STATUS		
	23		52	E8	003DC	BLBS	STATUS, 36\$		0665
		00000000G	EF	DD	003DF	PUSHL	SEMCOD		0669
	7E	00000000G	EF	3C	003E5	MOVZWL	TEMP, -(SP)		
		00000000G	EF	DD	003EC	PUSHL	TEMP+4		
		00000000G	8F	DD	003F2	PUSHL	#RNF COF		
00000000G	EF		04	FB	003F8	CALLS	#4, ERME		
			00C1	31	003FF	BRW	43\$		0670
	52	00208001	8F	DO	00402	MOVL	#2129921, STATUS		0675
00209000	8F		52	D1	00409	CMPL	STATUS, #2134016		0676
			25	13	00410	BEQL	38\$		
	50		67	DO	00412	MOVL	FOOIOB, R0		0681
34	A0		10	BO	00415	MOVW	#16, 52(R0)		
36	A0		02	90	00419	MOVB	#2, 54(R0)		
2C	A0		06	90	0041D	MOVB	#6, 44(R0)		
			59	DD	00421	PUSHL	R9		
			7E	D4	00423	CLRL	-(SP)		
			50	DD	00425	PUSHL	R0		

		6B	03	FB	00427	CALLS	#3, XPOSGET		
		52	50	DO	0042A	MOVL	R0, STATUS		
		OF	52	E8	0042D	BLBS	STATUS, 40\$	0683	
	00209000	8F	52	D1	00430	CMPL	STATUS, #2134016	0687	
			03	12	00437	BNEQ	39\$		
			008B	31	00439	BRW	44\$		
			FDA9	31	0043C	BRW	21\$	0694	
		50	EF	DO	0043F	MOVL	FOTIOB, R0	0699	
51		67	34	C1	00446	ADDL3	#52, FOOTIOB, R1		
	44	A0	51	DO	0044A	MOVL	R1, 68(R0)		
	2C	A0	07	90	0044E	MOVB	#7, 44(R0)		
			00000000G	EF	9F	PUSHAB	XPOSFAILURE		
				7E	D4	CLRL	-(SP)		
			50	DD	0045A	PUSHL	R0		
		6A	03	FB	0045C	CALLS	#3, XPOSPUT		
		52	50	DO	0045F	MOVL	R0, STATUS		
		4D	52	E9	00462	BLBC	STATUS, 41\$	0702	
		50	67	DO	00465	MOVL	FOOTIOB, R0	0713	
		51	A0	DO	00468	MOVL	56(R0), R1	0718	
34	A0	08	04	A5	0046C	MULW3	#4, 8(R1), 52(R0)		
		36	02	90	00472	MOVB	#2, 54(R0)		
	2C	A0	06	90	00476	MOVB	#6, 44(R0)		
			59	DD	0047A	PUSHL	R9		
			7E	D4	0047C	CLRL	-(SP)		
			50	DD	0047E	PUSHL	R0		
		6B	03	FB	00480	CALLS	#3, XPOSGET		
		52	50	DO	00483	MOVL	R0, STATUS		
		B3	52	E9	00486	BLBC	STATUS, 39\$	0721	
		50	EF	DO	00489	MOVL	FOTIOB, R0	0731	
51		67	34	C1	00490	ADDL3	#52, FOOTIOB, R1		
	44	A0	51	DO	00494	MOVL	R1, 68(R0)		
	2C	A0	07	90	00498	MOVB	#7, 44(R0)		
			00000000G	EF	9F	PUSHAB	XPOSFAILURE		
				7E	D4	CLRL	-(SP)		
			50	DD	004A4	PUSHL	R0		
		6A	03	FB	004A6	CALLS	#3, XPOSPUT		
		52	50	DO	004A9	MOVL	R0, STATUS		
		03	52	E9	004AC	BLBC	STATUS, 41\$	0734	
			FF57	31	004AF	BRW	37\$		
			7E	D4	004B2	CLRL	-(SP)	0738	
			52	DD	004B4	PUSHL	STATUS		
			8F	DD	004B6	PUSHL	#RNFCWF		
	00000000G	EF	03	FB	004BC	CALLS	#3, ERM	0739	
		50	03	DO	004C3	MOVL	#3, R0		
				04	004C6	RET		0748	
		50	67	DO	004C7	MOVL	FOOTIOB, R0		
	2E	A0	10	88	004CA	BISB2	#16, 46(R0)		
	2C	A0	02	90	004CE	MOVB	#2, 44(R0)		
			00000000G	EF	9F	PUSHAB	XPOSFAILURE		
				7E	D4	CLRL	-(SP)		
			50	DD	004DA	PUSHL	R0		
	00000000G	EF	03	FB	004DC	CALLS	#3, XPOS\$CLOSE		
		50	67	DO	004E3	MOVL	FOOTIOB, R0	0750	
	2C	A0	03	90	004E6	MOVB	#3, 44(R0)		
			7E	7C	004EA	CLRQ	-(SP)		
			50	DD	004EC	PUSHL	R0		
	00000000G	EF	03	FB	004EE	CALLS	#3, XPOS\$DELETE		

```

;      BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:FOOFIL/OBJ=OBJ$:FOOFIL MSRC$:FOOFIL/UPDATE=(ENHS$:FOOFIL)
; Size:          1343 code + 80 data bytes
; Run Time:      00:43.9
; Elapsed Time:  02:00.8

```

FOOFIL
V04-000

File operations for footnote processing
body of routine

M 11
16-Sep-1984 00:31:31

VAX-11 Bliss-32 V4.0-742

Page 22

; Lines/CPU Min: 1067
; Lexemes/CPU-Min: 53375
; Memory Used: 486 pages
; Compilation Complete

FOO
V04

0341

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY